

# DEFECT-TOLERANT FPGA ARCHITECTURE EXPLORATION

*Pongstorn Maidee and Kia Bazargan*

Department of Electrical and Computer Engineering  
University of Minnesota, USA  
email: pongstor,kia@ece.umn.edu

## ABSTRACT

According to the ITRS predictions, controlling manufacturing yield is going to be a challenging task in future technologies. The effective yield of future FPGA architectures considering configurable logic blocks, switch boxes, connection boxes and routing segments is estimated in this paper. The results show that some degree of redundancy for logic blocks, routing and switch boxes is necessary. However, no more than one spare logic block per cluster, and at most one spare wire is required to obtain a satisfactory effective yield. The results also indicate that it is beneficial to increase logic cluster size of future FPGA architectures for better yield.

## 1. INTRODUCTION

As feature sizes scale down, more and more defects appear as catastrophic faults which reduce the effective chip yield. To tolerate some faults, FPGAs must have spare or unused resources. However, the degree of redundancy depends on how comprehensively the chip is supposed to be tested for robustness. In a *full test* approach, flawless functionality of all resources is tested, and if some components are faulty, a defect map is created. On the other hand, in the *partial test* approach each chip is tested to see if a specific circuit can be mapped correctly [1, 2]. Therefore, the test pattern used is a subset of that of the full test approach. As a result, effective chip yield for those circuits are higher.

The defect map can be used to individually configure each chip to implement a circuit around the defects [3]. It is obvious that this approach is not scalable because generating the configuration for each chip is time consuming and not practical. However, if a replacement scheme is adopted during architecture design, spare resources can replace faulty ones locally based on the defect map. Therefore, the same configuration of the design can be used, with minimal changes which could be done automatically on chip during configuration. In this work, we assume such a replacement scheme<sup>1</sup>

Redundancy schemes for logic blocks, switch boxes, connection boxes, interconnects, IO buffers and configuration circuits have been proposed [4, 5, 6]. Expected yields for future generations of chips can be estimated based on the ITRS road map [1, 7]. It was shown that production yield will be increasingly lower and increasing cost in the future technologies.

Evaluating redundancy schemes for individual parts of an FPGA in isolation does not give us a realistic picture of the defect tolerance of the whole architecture. Interactions between defect tolerance of the parts, together with architectural parameters such as cluster size must be evaluated using a unified model for a realistic prediction.

The rest of the paper is organized as follows: the FPGA architecture used in this work is summarized in Section 2. Yield estimation and improvement are elaborated in sections 3 and 4 respectively. Yield estimation for the FPGA architecture are discussed in Section 5. The predictive yields for future technologies are also shown in this section as well as effects of cluster sizes on future yields. Finally, the conclusion is given in Section 6.

## 2. THE FPGA ARCHITECTURE MODEL

The architecture is shown in Figure 1. An FPGA consists of a two dimensional array of configurable logic blocks (CLB), switch box (SB), connection boxes (CB) and IO buffers. Switch boxes are connected to wires of different lengths: one, two, six and long wires that span the whole width or height of the chip [8]. We assume switch boxes are of the subset type, and one implication is that a wire can connect only to wires of the same type. For a given segment type, the ratio of the number of switch boxes that provide a connection to the segment to the total number of switchboxes along the length of the wire is denoted by  $F_s$ . We assume  $F_s = 1$  for all segment types except segments of length 6

approaches complement each other in that the former is intended for general users such as small volume and prototyping products, while the latter is for volume productions. They may also share the same architecture. Therefore, an FPGA can employ both schemes for its yield improvement. In such an FPGA, an appropriate redundancy scheme is decided and implemented.

---

<sup>1</sup>It is important to note that the redundancy scheme and the partial test

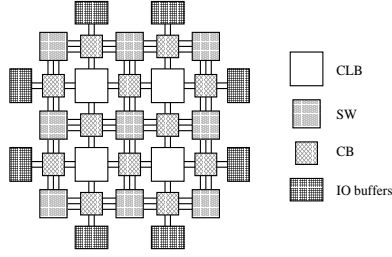


Fig. 1. An FPGA of size 2x2.

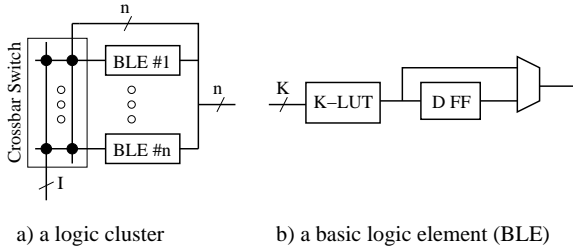


Fig. 2. CLB details.

which have  $F_s = 0.5$ , i.e., a wire of length 6 can be connected at its ends and at the middle. CLBs contain look-up tables that can be programmed to implement user logic functions. CLBs are connected to wire segments through CBs. The portion of CBs that can connect to passing wire segments is specified by  $F_c$ . We assume  $F_c = 1$  except a segment of length 6 which has  $F_c = 0.5$  in our study which means that a wire can connect to any CB it passes through. IO buffers, used to connect to off-chip circuits, are connected to wire segments in the same fashion as CLBs. A CLB consists of a cluster of basic logic elements (BLE) [9]. Their input/output pins are connected to every one of the wires from CBs. Each BLE contains one  $K$ -input look-up table (K-LUT), one D flip-flop and one multiplexer as shown in Figure 2. BLE output can be either latched or not. Let  $N$  be the number of BLEs in a cluster and  $I$  be the number of wires from CBs to the cluster. It is shown that the maximum BLE utilization is achieved when  $I = (N + 1)K/2$  [9]. It is also shown that  $K = 4$  requires small area for different cluster sizes with near optimal area-delay product. Therefore, we assume  $K = 4$  throughout this work.

### 3. YIELD ESTIMATION

The number of faults depend on both circuit structures and defect sizes. Defect size can be described as [10]

$$\begin{aligned} f_d(x) &= \frac{2x_0^2 x_M^2}{(x_M^2 - x_0^2)x^3} \quad \text{if } x_0 \leq x \leq x_M \\ &= 0 \quad \text{otherwise.} \end{aligned} \quad (1)$$

where  $x_0$  and  $x_M$  are the minimum and maximum defect sizes respectively. For a given defect size  $x$ , the area in which defects of type  $i$  will cause faults if they fall into is called the critical area  $A_i^{(c)}(x)$ . The effective critical area can be found by

$$A_i^{(c)} = \int_{x_0}^{x_M} A_i^{(c)}(x) f_d(x) dx. \quad (2)$$

Let  $\lambda$  and  $d_i$  be the average number of faults on the chip and of defects of type  $i$  per unit area, respectively. We have  $\lambda = \sum A_i^{(c)} d_i$ . Assuming infinite independent subregions, we obtain the Poisson distribution of random variable  $X$  [10]

$$P\{X = k\} = \frac{e^{-\lambda} \lambda^k}{k!} \quad (3)$$

where  $k$  is a constant denoting the number of defects. Therefore, the chip yield is

$$P\{X = 0\} = e^{-\lambda} = \prod_i \exp(-A\theta_i d_i). \quad (4)$$

To capture fault clustering, we represent  $\lambda$  as a gamma distribution  $f(l)$  with parameters  $(\alpha, \alpha/\lambda)$ . Integrating over possible defect sizes, we have a negative binomial yield formula as follows [10]

$$P\{X = k\} = \frac{\gamma(\alpha + k)}{k! \gamma(\alpha)} \cdot \frac{(\lambda/\alpha)^k}{(1 + \lambda/\alpha)^{\alpha+k}} \quad (5)$$

Note that  $\alpha$  ranges from 0.3 to 5 in practice. ITRS uses  $\alpha = 2$ . Thus, the chip yield can be computed by

$$P\{X = 0\} = (1 + \lambda/\alpha)^{-\alpha} \quad (6)$$

It is important to note that yield computation can be decomposed into different independent layers and different independent subareas.

## 4. YIELD IMPROVEMENT THROUGH REDUNDANCY

Different components in FPGAs such as CLBs and SWs are different in nature. Therefore, they require different replacement schemes. In this section, yield improvement by redundancy will be calculated for major FPGA components individually, but the simulations will simultaneously consider all these components.

### 4.1. Configurable Logic Block Redundancy

It has been shown that using local replacement schemes provide sufficient yield improvement, while suffering the least performance degradation. Inside clusters, a crossbar is used for local routing as shown in Figure 2 [9]. Therefore, inside a cluster, any BLE can serve as a redundant part. In

redundancy based clustering schemes, each cluster contains  $N + R$  logic blocks, in which  $R$  of them are spare. Therefore, at least  $N$  logic blocks must be fault free to make the cluster usable. The probability that  $m$  BLEs of a cluster are fault free is

$$y_m = \left(1 + m \frac{\lambda^b}{\alpha}\right)^{-\alpha} \quad (7)$$

where  $\lambda^b$  is the average number of faults for one logic block. Let the probability that exactly  $m$  out of  $M$  modules are working be  $F_m^M$ . By the Inclusion-Exclusion principle [10],

$$F_m^M = \binom{M}{m} \sum_{k=0}^{M-m} (-1)^k \binom{M-m}{k} y_{m+k}, \quad (8)$$

Therefore, the yield of a cluster is

$$Y = \sum_{i=N}^{N+R} F_i^{N+R} (-1)^j \binom{N+R}{i} \binom{N+R-i}{j} y_{i+j} \quad (9)$$

Note that connection box redundancy can be computed similar to CLBs as will be seen in Section 5.1.

## 4.2. Routing Redundancy

In general, FPGAs are implemented in a tile-based approach, in which one tile containing one logic cluster, two connection boxes, one switch box and routing channels is designed. Therefore, it is reasonable to assume that segmented wires are laid out as parallel wires with minimum width and spacing specified by the technology information. Therefore, a yield estimation similar to the one proposed in [11] can be used.<sup>2</sup>

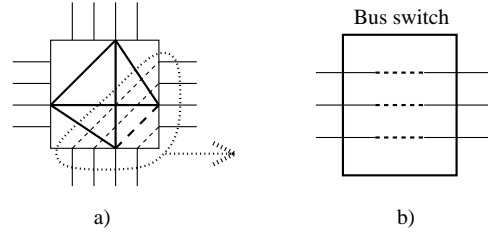
Let  $n$  be the number of wires in one channel in the original retardant-free architecture. Let  $r$  and  $T$  be the number of additional wires and of working wires on the bus, respectively. The probability that this bus will work is [11]

$$P(T \geq n) = \sum_{k=n}^{n+r} (-1)^{k-n} \binom{k-1}{n-1} W(k) \quad (10)$$

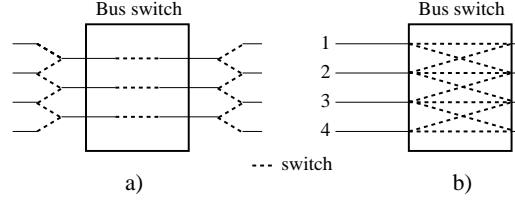
where  $W(k)$  is the sum of the probability that a subset of size exactly  $k$  is working. For a given set of wires, its probability depends on the area it covers and the spacing among them. For any  $k$  wires, we can define the number of groups of consecutive working wires,  $g$ . Thus, the probability that this set of  $k$  wires works is

$$P(k, r) = \left[1 + \frac{d_1}{\alpha_1} \theta_1 k(w+s)L\right]^{-\alpha_1} \cdot \left[1 + \frac{d_2}{\alpha_2} \theta_2 (k+r)(w+s)L\right]^{-\alpha_2} \quad (11)$$

<sup>2</sup>We have not modeled the staggering of the wire segments of length 2 and 6, but believe that the change in the layout model to account for the staggering of lines minimally affects our yield models.



**Fig. 3.** A switch box and its decomposition.



**Fig. 4.** Bus switch redundancy of 2 buses with  $n = 3, r = 1$ .

where  $\theta_1 = \frac{x_0^2}{w(2w+s)}$ ,  $\theta_2 = \frac{x_0^2}{s(2s+w)}$  and  $d_1$  and  $d_2$  are defect densities of open and short wires respectively. For a bus of  $n + r$  wires, the number of subsets of size  $k$  that are divided into  $g$  groups of consecutive working wires is

$$R_g^{(n+r,k)} = \binom{n+r-k+1}{g} \binom{k-1}{k-g} \quad (12)$$

Therefore, we have

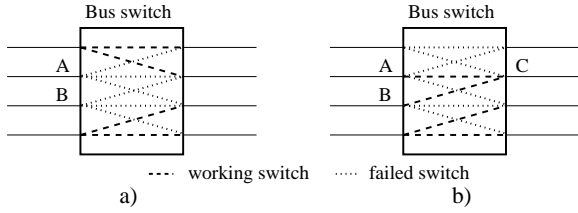
$$W(k) = \sum_{g=1}^k R_g^{(n+r,k)} P(k, g) \quad (13)$$

## 4.3. Switch Box Redundancy

Failure within one switch box will affect the usage of other switch boxes because they are connected in a mesh fashion. Determining the yield of a switch box mesh is NP-hard. Therefore, in this work we consider each switch box in isolation. Since a pair of wires from different sides of a switch box can be connected independently, we decompose a switch box, as shown in Figure 3-a, into 6 independent *bus switches*, one of which is shown in Figure 3-b.

Let us consider two buses of  $n$  wires and  $r$  additional redundant wires connecting together through a bus switch as shown in Figure 4-a. Even though buses are defect tolerant, the bus switch is susceptible to defects. A defect tolerant bus switch is shown in Figure 4-b. A wire in the middle of the bus requires  $2r + 1$  switches, but wires near the border need fewer switches. The total number of switches in the defect tolerant bus switch is  $(n - r)(2r + 1) + 2 \sum_{i=0}^{r-1} (2r - i)$ .

We define the yield of the bus switch as the probability that at least  $n$  wires can be connected to  $n$  distinct wires on the other side. Let us consider Figure 5. Both wires  $A$  and  $B$  will fail if all switches connect to them fail, as shown in Figure 5-a. However, even if not all of them fail, they may be



**Fig. 5.** Failure of bus switch with  $n = 3, r = 1$ .

forced to share the same target segment, as shown in Figure 5-b, which translates into failure. We will consider only the first case as failure.

Let  $p$  be the probability that a switch in the bus switch fails. Assuming that each switch fails independently, the probability that a wire will fail is  $p^{(2r+1)}$ . By using the Inclusion-Exclusion principle, the yield of a bus switch can be computed as

$$\text{prob}\{\#\text{wire fail} \leq r\} = \sum_{i=0}^r \sum_{k=0}^{n+r-i} (-1)^k \binom{n+r}{i} \binom{n+r-i}{k} (p^{2r+1})^{i+k} \quad (14)$$

## 5. RESULTS

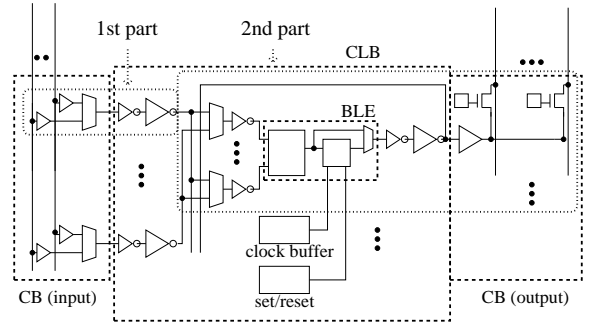
In this section, yield computation of an architecture under study will be elaborated in Section 5.1. Yield prediction results for future FPGA architectures with varying degrees of redundancy are presented in Section 5.2. The effect of cluster sizes on the yield is investigated in Section 5.3.

### 5.1. FPGA Yield Computation

The fault density of a given circuit depends on its layout. Since interconnect layout is fairly simple, the interconnect yield can be computed with high accuracy. However, logic cluster layouts are complicated. Therefore, we resort to a layout independent approach which allows us to adjust LUT yields for different cluster sizes to provide a fair initial yield at the current technology.<sup>3</sup>

According to ITRS [12], the yield due to random defects is between 83-89.5%. The LUT yield of each architecture at current technology is assumed to be given. For any given channel width (CW), the number of segments of length 1, 2, 6 and long wires are 0.08, 0.2, 0.6 and 0.12, respectively [8]. However, the number of segment of length 1, 2 and 6 starting at a SW denoted  $n1, n2, n6$  is  $0.08CW, 0.2CW/2$ , and  $0.6CW/6$  respectively. We assume that each type of segments will have the same number of redundant wires denoted by  $r$ . Let the cluster size of a given architecture be  $N$  and the number of spare BLEs be  $R$ . Therefore, the number of inputs to a cluster should be  $I = 2(N + R + 1)$ .

<sup>3</sup>It is important to note that for any given LUT layout, its yield can be computed and the methodology used here can be applied.



**Fig. 6.** Details of a configurable logic box and its connection box.

A cluster contains BLEs, FFs, MUXs, buffers as well as clock buffer and set / reset logic as shown in Figure 6 [13]. We assume that global parts, *i.e.*, clock buffer and set/reset logic, are fault free. We measure circuit area in terms of the number of minimum size transistors,  $tr_{min}$ . For a given number of BLEs and a number of inputs to a cluster, the CLB area can be computed. Inputs to a CLB, part of CB, are implemented using multiplexers with  $m$  inputs requiring  $(6\lceil\log(m)\rceil + 2m - 2) tr_{min}$ , including its SRAMs [13]. Inputs to the multiplexer are driven by buffers shared by 2 CLBs on both sides of the wires. Each buffer requires  $9.25 tr_{min}$ . The other part of a CB is used to connect CLB outputs to routing resources using shared 16x buffers, each of size  $39.9 tr_{min}$  and pass transistors each of size  $11.5 tr_{min}$  including its controlling SRAM [13]. Therefore, the area of a CB connecting a cluster of size  $N + R$  to a bus of  $m$  wires is  $(6\lceil\log(m)\rceil + 2m - 2) \times I + 9.25m/2 + 39.9(N + R) + 11.5m(N + R) tr_{min}$ .

Each segment emanating from a switch box requires 6 switches. For a given redundancy  $r$ , a wire will connect to  $2r + 1$  wires. A long wire can connect to another perpendicular wire at any SBs. But, segments of length 2 and 6 can connect only at the middle. Therefore, the total number of switches at a switch box is  $6(2r+1)(n1+r) + 6(2r+1)(n2+r) + 6(2r+1)(n6+r) + (n2+r) + (n6+r) + (nL+r)$ .

A complete tile of a four 4-LUT cluster takes  $25983\mu m^2$  in a  $0.18\mu m$  technology. It is linearly scaled down to the  $0.078\mu m$  for a 2006 or future technology nodes. Since one 4-LUT requires  $1801 tr_{min}$  [13], the area of  $1 tr_{min}$  can be estimated and used to compute CB or SB areas. A tile's active area can be computed for future technologies using array sizes from Table 1. Segments of different types and their horizontal / vertical orientations are assumed to be implemented in different metal layers. We also further assume that segments are laid out using the minimum intermediate wire pitch. Therefore, a total area of wires of different type and orientation can be computed separately. Finally, chip area can be obtained from the maximum of active and wiring areas.

For a cluster of  $N$  BLEs with  $R$  spare BLEs. Only  $2(N +$

**Table 1.** FPGA array parameters (see Section 5.2 for its calculation).

parameter \ cluster size	1	2	3	4	5	6
Array size (2006)	92	65	53	46	41	38
Channel width	32	44	50	59	63	70
parameter \ cluster size	7	8	9	10	11	12
Array size (2006)	35	33	31	29	28	27
Channel width	75	83	87	91	97	101

1) inputs and  $N$  outputs out of  $2(N + R + 1)$  inputs and  $N + R$  output are required. For a given LUT yield,  $\lambda^b$  can be computed by (7). We assume that the layout are uniform<sup>4</sup>. Hence,  $\lambda$  of an CB input and an BLE with its output, shown in the small-dash areas, can be computed proportionally to their area. A cluster yield can be computed in two parts: 1) CB inputs and 2) BLE and its CB output, shown in Figure 6, each by using (9). Finally, the cluster yield is their product.

Since we assume a subset switch box, we can consider each segment type and its associated switches separately. Since a tile width is known, physical lengths of each type of segments can be computed and used to compute their yields as mentioned in Section 4.2.

A switch box can be decomposed into 6 independent bus switches. Consider a bus switch shown in Figure 3b. Let  $Y_{sw}^r(n_i)$  be a yield of connecting all  $n_i$  wires from one side to the other computed by (14). Based on the discussion above, the yield of a switch box in our architecture is

$$Y_{sw}^r(n_1)^6 Y_{sw}^r(n_2)^7 Y_{sw}^r(n_6)^7 Y_{sw}^r(n_L). \quad (15)$$

Finally, the chip yield can be computed as the product of all its component yields.

Since adding redundancy will increase the chip size, there is less number of chips per wafer. Therefore, to take extra area into account, we define the effective yield as

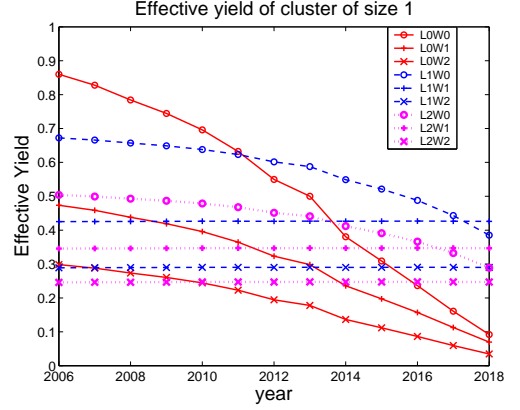
$$Y_{eff} = Y_r \frac{N_r^R(H_r, W_r^R)}{N_0^0(H_0^0, W_0^0)}, \quad N(H, W) = \frac{\pi R_e^2}{HW} e^{-\frac{H}{R_e}} \quad (16)$$

,where  $H, W$  are height and width of the chip,  $R_e$  is the wafer radius [7]. The ratio  $N_r^R(H_r, W_r^R)/N_0^0(H_0^0, W_0^0)$  will always be less than 1 reflecting area overhead.

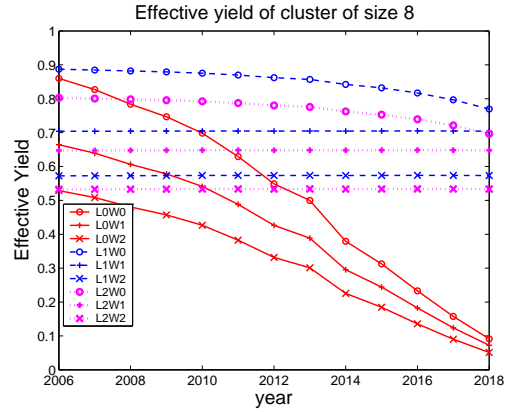
## 5.2. Future Yields of the Current Architectures

In this work, the Toronto20 benchmarks are used to determine the size of the FPGA architecture. Since *clma* contains the largest number of gates and *pdc* requires the largest channel width, they are used to determine, through the VPR tool, the array sizes and routing resources with different cluster sizes whose values are shown in Table 1. For future FPGAs of a given cluster size, we assume that the array size

<sup>4</sup>In practice, using the layout from previous generation FPGAs, critical area of each part can be computed and used in the analysis.



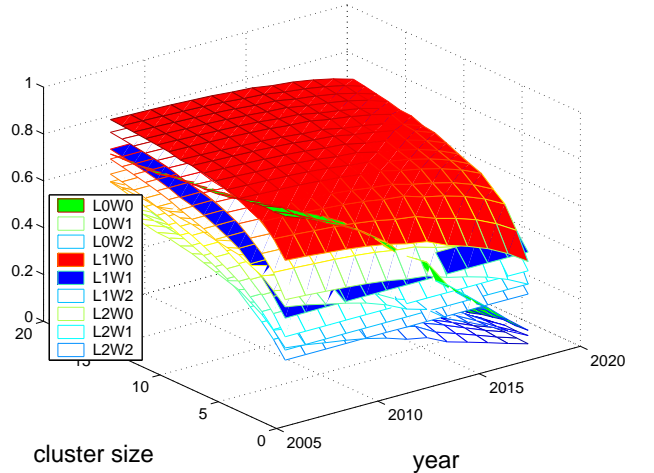
**Fig. 7.** Effective yield of cluster of size 1.



**Fig. 8.** Effective yield of cluster of size 8.

increases linearly due to the reduction in gate length. However, we assume that the routing complexity of circuits remains the same, making the channel width constant.

For any cluster size, LUT yield is set so that the yield at current technology is 86.0%. Since both layout geometries and critical defect size scale down at the same rate for future technologies, LUT yield remains constant according to (2). The predictive effective yields of different cluster sizes were



**Fig. 9.** Effective yields using fixed LUT yield.

studied, some of them were plotted in Figure 7-8. Different redundancies denote by  $LR$  and  $r$ , where  $R$  and  $r$  is the number of spare BLEs per cluster and spare wires for each segment type, respectively.

For small cluster size as in Figure 7,  $L0W0$  provides the best yield in 2006. In the future  $L1W0$  dominant because  $R = 1$  provides redundancy. Eventually, once the wire yield is very low,  $r = 1$  is needed. However, when cluster size increases, as shown in Figure 8,  $R = 1$  does not incur too much overhead but provides yield improvement. Therefore,  $L1W0$  provides better yield over the period of consideration, even at the current technology.

Note that the  $LRWr$ , for  $R, r > 0$ , remains almost constant in both Figure 7 and 8 because their yields are almost 100% and their area overhead comparing to the nonredundant architecture remain constant. The  $LRWr$ , for  $R, r > 1$  give a little bit better yield than  $L1W1$  but incur more area overhead resulting in less effective yields than that of  $L1W1$ .

Providing redundancy for all interconnect channels incurs area overhead for wiring itself as well as for SBs. Therefore, the area overhead outweigh the before-effective yield improvement except only for the very low interconnect yield. As a result, in spite of the fact that interconnect comprises large portion of an FPGA,  $L1W0$  outperforms  $L1W1$  in Figure 7 and most of the period in Figure 8.

### 5.3. Effect of Cluster Size on Future Yield

In this study, we envision the situation that an architecture was designed and implemented for one specific cluster size. However, we would like to see how chip yields change if we decide to change the cluster size as well as its associated parameters, shown in Table 1. Results are shown in Figure 9. We assume that the same LUT layout will be used in the architecture variations. The result shows that  $L1W0$  provides the best yield except for near future with small cluster sizes. Considering  $L1W0$ , at any particular year, the effective yield monotonically increases with cluster size. Therefore, while an architecture with small cluster size gives the best yield at the present, it is beneficial to increase the cluster sizes in the future.

## 6. CONCLUSION

The comprehensive effective yield of FPGA considering configurable logic blocks, switch boxes, connection boxes and routing segments is estimated in this paper. Using VPR tool to define equivalent architectures for different cluster sizes, we can compare effective yields of different cluster sizes. The results show that only one spare logic block per cluster and at most one spare wire per channel of each segment type are enough for future technologies. Furthermore, a cluster size affects future FPGA architectures in two ways: 1) for large cluster size, only one spare BLE is needed, but not spare wires. 2) effective yields increase with cluster size. Combining larger cluster size and redundancy can give a satisfactory effective yield for future technologies.

The routing resources yield is derived assuming minimum wire pitch. In practice, wires are spaced wider to avoid crosstalk noise. As a result, the practical routing resource yield would be higher than estimated. As the yield increases, the redundancy benefit decreases because its area overhead is constant. Therefore, the spare wire will not be required in practice for large cluster sizes. However, the effective yield of  $LRW0$  would be higher. The switch box yield is overestimated since some switch failing cases are ignored. Therefore,  $LRW0$  is optimistic. Since  $LRWr$ , for  $r > 0$  is high, there is not much change in their yield. As a result, the gap between  $LRW0$  and  $LRWr$ , for  $r > 0$  is smaller. However, the global trends still remain the same.

## 7. REFERENCES

- [1] N. Campregher, P. Y. K. Cheung, G. A. Constantinides, and M. Vasilko, "Yield enhancements of design-specific FPGAs," in *Proc. of the Int. Sym. on FPGA*, 2006, pp. 93–100.
- [2] Xilinx, "Easypath solutions," 2006. [Online]. Available: <http://www.xilinx.com>
- [3] V. Lakamraju and R. Tessier, "Tolerating operational faults in cluster-based FPGAs," in *Int. Sym. on Field programmable gate arrays*, 2000, pp. 187–194.
- [4] N. Howard, A. Tyrrell, and N. Allinson, "The yield enhancement of field-programmable gate arrays," *IEEE Trans. VLSI Syst.*, vol. 2, no. 1, pp. 115–123, 1994.
- [5] Altera, "Apex redundancy," 2004. [Online]. Available: <http://www.altera.com>
- [6] F. Hanchek and S. Dutt, "Methodologies for tolerating cell and interconnect faults in FPGAs," *IEEE Trans. Comput.*, vol. 47, no. 1, pp. 15–33, 1998.
- [7] N. Campregher, P. Y. K. Cheung, G. A. Constantinides, and M. Vasilko, "Analysis of yield loss due to random photolithographic defects in the interconnect structure of FPGAs," in *Proc. Int. Sym. on FPGA*, 2005, pp. 138–148.
- [8] Xilinx, "Virtex-II data sheets," 2005. [Online]. Available: <http://www.xilinx.com/>
- [9] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deep-submicron FPGA performance and density," *IEEE Trans. VLSI Syst.*
- [10] I. Koren and Z. Koren, "Defect tolerance in VLSI circuits: techniques and yield analysis," *Proc. IEEE*, vol. 86, no. 9, pp. 1819–1838, 1998.
- [11] I. Koren, Z. Koren, and D. Pradhan, "Designing interconnection buses in VLSI and WSI for maximum yield and minimum delay," *IEEE J. Solid-State Circuits*, vol. 23, no. 3, pp. 859–866, 1988.
- [12] Semiconductor Industry Association, "The International Technology Roadmap for Semiconductors," 2005. [Online]. Available: <http://public.itrs.net>
- [13] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-submicron FPGAs*. Kluwer Academic Pub., 1999.